Comparing user-dependent and user-independent training of CNN for SSVEP BCI

To cite this article before publication: Aravind Ravi *et al* 2020 *J. Neural Eng.* in press https://doi.org/10.1088/1741-2552/ab6a67

IOP Publishing Journal Of Neural Engineering
Journal **XX** (XXXX) XXXXXX https://doi.org/XXXX/XXXX

Comparing User-Dependent and User-Independent Training of CNN for SSVEP BCI

Aravind Ravi[1], Nargess Heydari Beni[1], Jacob Manuel[1] and Ning Jiang[1]

[1] Department of Systems Design Engineering, University of Waterloo, Waterloo, ON, Canada

E-mail: ning.jiang@uwaterloo.ca

Received xxxxxx
Accepted for publication xxxxxx
Published xxxxxx

Abstract

Objective. We presented a comparative study on the training methodologies of Convolutional Neural Network (CNN) for detection of steady-state visual evoked potentials (SSVEP). Two training scenarios were also compared: user-independent (UI) training and user-dependent (UD) training. *Approach.* The CNN was trained in both UD and UI scenarios on two types of features for SSVEP classification: magnitude spectrum features (M-CNN) and complex spectrum features (C-CNN). And the Canonical Correlation Analysis (CCA), widely used in SSVEP processing, was used as the baseline. Additional comparisons were performed with Task-Related Components Analysis (TRCA) and Filter-bank Canonical Correlation Analysis (FBCCA). The performance of the proposed CNN pipelines, CCA, FBCCA and TRCA were evaluated with two datasets: a seven-class SSVEP dataset collected on 21 healthy participants and a twelve-class publicly available SSVEP dataset collected on 10 healthy participants. *Main results.* The UD based training methods consistently outperformed the UI methods when all other conditions were the same, as one would expect. However, the proposed UI-C-CNN approach performed similar to the UD-M-CNN across all cases investigated on both datasets. On Dataset 1, the average accuracies of the different methods for 1 s window length were: CCA: 69.1±10.8%, TRCA: 13.4±1.5%, FBCCA: 64.8±15.6%, UI-M-CNN: 73.5±16.1%, UI-C-CNN: 81.6±12.3%, UD-M-CNN: 87.8±7.6% and UD-C-CNN: 92.5±5%. On Dataset 2, the average accuracies of the different methods for data length of 1 s were: UD-C-CNN: 92.33±11.1%, UD-M-CNN: 82.77±16.7%, UI-C-CNN: 81.6±18%, UI-M-CNN: 70.5±22%, FBCCA: 67.1±21%, CCA: 62.7±21.5%, TRCA: 40.4±14%. Using t-SNE, visualizing the features extracted by the CNN pipelines further revealed that the C-CNN method likely learned both the amplitude and phase related information from the SSVEP data for classification, resulting in superior performance than the M-CNN methods. The results suggested that UI-C-CNN method proposed in this study offers a good balance between performance and cost of training data. *Significance.* The proposed C-CNN based method is a suitable candidate for SSVEP-based BCIs and provides an improved performance in both UD and UI training scenarios.

Keywords: brain-computer interface, steady-state visual evoked potential, electroencephalography, convolutional neural networks, user-independent, user-dependent, calibration-free

1. Introduction

Brain-Computer Interfaces (BCIs) offer a direct communication path between the brain and a computer that allows control of an external device without the need of the conventional neuromuscular system [1]. BCIs detect changes in the physiological signals recorded from the brain that are associated with users' mental state or intentions and then translate them into useful commands. BCIs provide novel

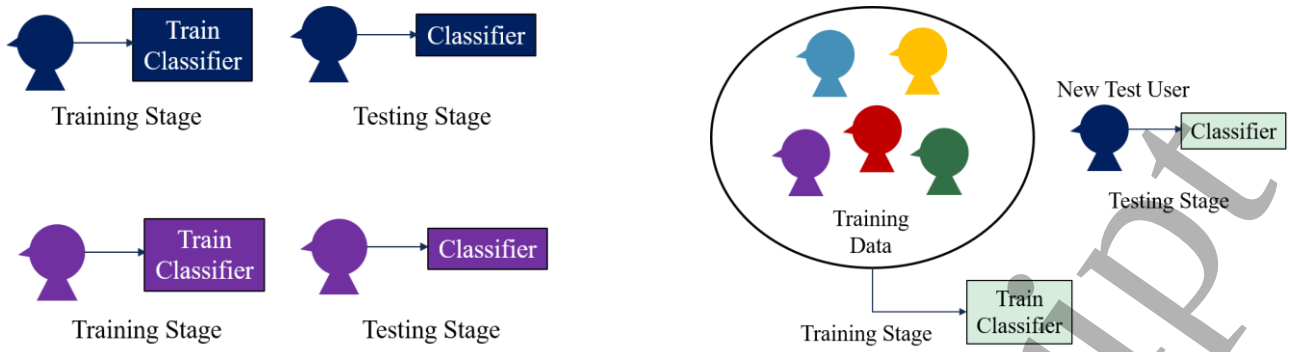


Figure 1. A diagram representing the User-Dependent (UD) and User-Independent (UI) training scenarios

possibilities for the neurorehabilitation for people with neurological disease such as stroke, amyotrophic lateral sclerosis (ALS) or paralysis [2], [3], [4]. The most common non-invasive method of recording brain activity for BCIs is based on electroencephalogram (EEG), through which it is possible to measure the changes in the neuronal activities that are related to mental state and/or intentions of the BCI user and extract useful features that allow the user to interact/communicate with the external environment. BCIs can be classified into two broad categories: endogenous BCIs and exogenous BCIs [5]. Endogenous BCIs allow a user to modulate his/her neuronal activity completely based on covert intentions. These include paradigms such as motor imagery BCI, imagined tactile responses based BCI, etc. Exogenous BCIs are dependent on an external stimulus to modulate the user's neuronal activity, providing contextual information regarding the intention of users. Examples include P300 based BCI, steady-state visual evoked potentials (SSVEP) based BCIs, steady-state motion visual evoked potential (SSMVEP) BCIs, etc. A key advantage of exogenous BCIs over endogenous BCIs is that they do not require as much user training. In the present study, we investigate the SSVEP BCI paradigm. Some of the desirable properties that SSVEP based BCIs offer include low participant training time, high signal-to-noise ratio and high information transfer rate (ITR). In this paradigm, one or more visual stimuli in the form of flickering light sources are presented to the user on a computer screen with each stimulus flickering at a certain frequency. When the user attends to one of the stimuli, an SSVEP response is elicited in the occipito-parietal region of the brain. These responses manifest as an increase in the amplitude of the EEG at the corresponding flicker frequency and often its harmonics. Each stimulus is usually mapped to a command on the external control application. The stimulus with the user's attention can be identified by analyzing the dominant response in the EEG and the corresponding command is generated.

As with any human-machine interface, there are two learning agents in BCI: the user and the algorithm. One of the advantages of SSVEP-based BCIs is that little, if any at all, user training is required. On the other hand, several feature extraction and classification algorithms have been developed for SSVEP processing, which can be categorized into three broad categories: training-free methods, user-specific training methods and user-independent training methods [6]. Training-free algorithms do not require any training data and the user of the BCI can immediately start using the system [7], [8], [9], [10]. The most widely used training-free method for SSVEP classification is the Canonical Correlation Analysis (CCA) [7], [8]. This method has been used as the baseline algorithm for SSVEP detection. CCA is a multi-variable statistical technique that allows to capture the underlying correlation between two random variables. One variable is the EEG data and the other variable is a set of sinusoidal reference templates corresponding to the stimuli frequencies.

Secondly, the user-specific or user-dependent (UD) training methods require training data from each user, from which a user-specific model is generated. Finally, the user-independent (UI) training methods require training data from multiple participants and a generalized model, or a 'user-independent' model, is generated such that it can be applied to unseen users. This method is particularly suited for SSVEP BCIs as these responses are more consistent across most humans, compared with other signal modalities of BCI such as sensory-motor rhythm [11]. Figure 1 illustrates the UD and UI training methods. Among the three categories, the UI method is favorable next to the training-free method as this does not require any training/calibration data to be collected on a new unseen user, and thus making the system virtually calibration-free once properly trained. Most UD and UI approaches for SSVEP have been extensions of the classical CCA method that incorporate some form of template matching scheme in addition to the sinusoidal reference templates. Most widely used UD methods include: Combination method-CCA [12], Individual Template CCA (IT-CCA) [13] and more recently proposed Task Related Components Analysis (TRCA) [14]; UI methods include: Filter-Bank CCA (FBCCA) [15], Combined-tCCA and Adaptive Combined CCA (A3C) [16].

Recently, there has been an increased interest in the application of deep learning algorithms for detection and classification in EEG based BCI [17] [18], [19]. Deep learning offers the advantage of automatic feature extraction either in the time domain EEG or in the transform domain as opposed to sophisticated feature extraction methods. A recent survey indicated that 41% of studies used some form of transform/feature extraction before applying a deep neural

network [19]. The convolutional neural network (CNN) was the most prevalent among these studies accounting for 43% of the studies [19]. Many recent studies have shown that CNNs provide significant improvement in performance compared to traditional techniques for SSVEP detection [20]–[25]. Among these studies, [21], [23], [24] have transformed the SSVEP trials into the frequency domain before providing as input to the CNN for classification. In [24], an asynchronous SSMVEP BCI was developed in which EEG was converted to the frequency domain using a Fast Fourier Transform (FFT) and then applied as input to the CNN to distinguish between Intentional Control (IC) and No Control (NC) state. Subsequently, CCA was used to classify the SSMVEP targets. This approach was shown to outperform the traditional approaches such as CCA-Threshold and CCA-kNN methods. A similar FFT based transformation was applied to the SSVEP data in [21] and [23]. In [23], the authors showed that the CNN classification was better than LASSO [26] in decoding the SSVEP targets. It is important to note that these studies have used only the UD based training procedure. On the other hand, [22] was one of the early studies to evaluate a UI based training procedure for SSVEP detection using a CNN. The authors used the time domain EEG directly as input to the CNN and showed that it was able to learn discriminable features to classify twelve unique SSVEP targets. In addition to this, the authors showed the importance of phase related information present in the SSVEP data that aided in better classification accuracy. Although these studies have independently done UD and UI training of CNN, to our knowledge, there are limited number of studies comparing both UD and UI training of CNN for SSVEP detection. A recent survey on training methods for SSVEP BCIs indicated that there was a glaring gap in the literature for lack of comparative performance studies between UD and UI methods for SSVEP BCIs [6]. Therefore, in the current study, we address this gap by providing a comparison of UD and UI training methods for SSVEP BCIs. Specifically, we compare the performance of different feature extraction methods with CNN for SSVEP classification.

A generic architecture that works across multiple datasets is highly desirable for any deep neural network based approach. Using time domain as input to a CNN poses some challenges in this regard. The dimensions of the input data directly depend on the sampling rate of the EEG system. A subsequent up or down sampling step maybe required in a case where the CNN model trained on a lower sampling rate was to be used on a data with higher sampling rate, and this could lead to loss of information. The ITR is directly influenced by the window length of the time domain data, and thus require modifying the input layer of the CNN when window length changes. These challenges can be addressed when the frequency domain representation is used as input to the CNN and can be achieved by fixing the resolution of the FFT. Moreover, these earlier studies using CNN were exclusively based on the magnitude spectrum of FFT and had ignored the phase related information [21], [23]. Many studies have used frequency-phase coding approach in the stimulus design [12], [27] and use the phase information as part of the classifier to detect the SSVEP targets [28], [29]. Models that ignore the phase information could potentially underperform on these datasets. These models were computationally lighter models compared to the CNN using time-domain data [22], the goal of which was to modify a previously proposed Compact-CNN [30] for detection in BCI tasks such as P300, error-related negativity (ERN), movement related cortical potentials (MRCP) and sensorimotor-rhythms (SMR) to be re-purposed for the application of SSVEP BCIs. Although a generic architecture that works across multiple BCI paradigms is desired, a task-specific CNN model could provide high performance and simultaneously provide a less complex architecture.

In the current study, we propose to combine the real and imaginary parts of the FFT and provide as input to the CNN as this combines both the amplitude and phase related information in SSVEP for decoding the targets. The preliminary results of this approach was presented in [31]. The proposed method aimed to study the performance in UD and UI scenarios while maintaining a simple architecture. This architecture was inspired by [21] and was used in a previous study [32]. This model achieved reduced computational complexity and reduced number of tunable parameters compared to previously published CNN models in [22] for SSVEP classification. One of the key challenges highlighted in deep learning based methods for BCIs is reproducibility of results [18], where the authors provided guidelines such as: providing a clear description of the architecture, clearly describing the data used, use of existing datasets and evaluating the performance with baseline. Therefore, the proposed method was compared with the magnitude spectrum based transformation under both UD and UI training scenarios. CCA was used as the baseline algorithm. FBCCA and TRCA were also compared with the proposed methods. In addition, two datasets were used in this study for the comparison: 1) a seven class SSVEP dataset with 21 participants recorded in our lab and; 2) a publicly available twelve class SSVEP dataset with 10 participants, which used in many earlier studies [12], [13], [16], [22]. Additional comparisons were performed with other published methods that reported the classification accuracies on the same public dataset.

In the next section, the datasets and methodologies used in this study are detailed. In Section 3, the results of the comparison of all the methods on both seven class and twelve class datasets are presented. Section 4 discusses the results of the experiments. The conclusion and directions for future work are provided in Section 5.

2. Methodology

2.1 Dataset Description

The proposed methods and other previously published methods were evaluated and compared on two datasets; a dataset acquired in our lab – Dataset 1 [33] and a public dataset – Dataset 2 [12].

2.1.1 Dataset 1. Twenty-one healthy adults (6 Females and 15 Males, aged 19-28 years) with normal or corrected-to-normal vision participated in an offline experiment. The

experiment was approved by the Office of Research Ethics of the University of Waterloo (ORE # 31850). A written informed consent was signed by each participant before starting the experiment. All participants were seated in a comfortable chair at 0.6 meters from an LCD monitor. Seven flickering stimuli were displayed on the monitor (60 Hz refresh rate) with the following flicker frequencies: 8.423Hz, 9.375Hz, 9.961Hz, 10.84Hz, 11.87Hz, 13.4Hz, and 14.87Hz, respectively. One stimulus was fixated at the center of the screen and six surrounding stimuli were placed concentrically around the central stimulus. Each stimulus was white in colour and circular in shape as they were shown to elicit the better SSVEP responses than other shapes and colors [34], [35] . The inter-stimulus distance or viewing angle was measured as a function of the distance between the centers of each stimulus. This was fixed as 5.24^{o} as used in previous studies [33].

EEG data was acquired using the g.USBamp and Gammabox (g.tec Guger Technologies, Austria) wet electrode (g.Scarabeo) system with a sampling rate of 1200 Hz. Six active electrodes were placed at the occipital and occipito-parietal areas as follows: O1, O2, Oz, PO3, POz and PO4, according to the International 10-20 system. FPz was used as the ground and right ear lobe was used as the reference.

The experimental protocol was similar to the one used in our previous study [33]. At the beginning of each trial, the participant was directed by a visual cue (yellow marker above the target stimulus) to gaze at the target stimulus of the trial on the screen. This cuing period was 2s. The participant was asked to focus on the target stimulus for a 6-second period. A break of 4s between two consecutive trials was provided. Each stimulus was repeated eight times in a single run, resulting in 56 trials per run. A pseudorandom sequence was generated for stimulus presentation. In addition, the participants were asked to avoid any eye blinks or sudden jerky movements during the trials. The experimental protocol and stimulus were designed using OpenViBE software [36]. All data were recorded, stored and analyzed offline.

2.1.2 Dataset 2. An offline SSVEP dataset collected on ten healthy volunteers was downloaded from a public repository [12]. All participants were seated in a comfortable chair at 0.6 meters from an LCD monitor in a dim room. Twelve flickering stimuli were displayed on the monitor with the following flicker frequencies: 9.25Hz, 9.75Hz, 10.25Hz, 10.75Hz, 11.25Hz, 11.75Hz, 12.25Hz, 12.75Hz, 13.25Hz, 13.75Hz, 14.25Hz, and 14.75Hz. The stimuli were arranged in a 4x3 grid of 6cm x 6cm squares that represented a numeric keypad.

The EEG data was acquired using the BioSemi ActiveTwo EEG (Biosemi B.V., Netherlands) system with a sampling rate of 2048Hz. Eight active electrodes were placed over the occipito-parietal areas. At the beginning of each trial, the participant was directed by a red square cue to gaze at a specific stimulus on the screen. The cuing period was 1s. The participant was asked to focus on the targeted stimulus for a duration of 4s. One block consisted of 12 trials with one trial for each of the 12 stimuli on the screen. They were presented in a random order. A total of 15 blocks were presented leading to a total of 180 trials.

2.2 Pre-Processing and Feature Extraction

2.2.1 Pre-Processing. The pre-processing for each dataset was performed separately. For Dataset 1, the signals from three occipital channels O1, O2 and Oz were filtered using a 4th order Butterworth band-pass filter between 1Hz and 40Hz. Each 6s trial was then segmented with a sliding windows scheme with different widths: 0.5s, 1s, 1.5s, 2s, 2.5s, 3s, and with a step of 100ms to bootstrap the number of training epochs. For Dataset 2, the signals were pre-processed based on [12] and [22] for comparison. All eight channels were used from this dataset. Consistent with the analyzing method in [12] and [22], a 4th order Butterworth band-pass filter between 6Hz and 80Hz was used to filter the data. Each 4-second trial was divided into 1s non-overlapping segments as per [22].

2.2.2 Magnitude Spectrum Features. Prior studies have considered the use of magnitude spectrum features as input to a CNN for SSVEP classification [21], [23], [24], [32]. In these prior studies, the pre-processed time-domain EEG signals $x(n)$ were transformed into the frequency domain $X(k)$ by computing the FFT resulting in a sequence of complex numbers $Re(X(k)) + jIm(X(k))$, from which the magnitude spectrum was calculated: $|X(k)| = \sqrt{Re(X(k))^2 + Im(X(k))^2}$. In the current study, the frequency resolution of the FFT was fixed as 0.2930Hz and the frequency components between 3Hz and 35Hz were selected. As a result, the length of the FFT transformed signal was $N_{fc} = 110$. The resultant signal computed along each channel were stacked one below the other to form a matrix with dimensions $N_{ch} \times N_{fc}$, where N_{ch} was the number of channels and N_{fc} was the number of frequency components and provided as input to the CNN. In this study, we refer to this approach as the M-CNN method. An example of the input I_{M-CNN} for Dataset 1 is defined as:

$$I_{M-CNN} = \begin{bmatrix} |FFT(x_{O1})| \\ |FFT(x_{Oz})| \\ |FFT(x_{O2})| \end{bmatrix} \quad (1)$$

This approach only considers the magnitude at different frequencies, with the phase information ignored. Earlier studies have shown that phase information provides significant information in decoding SSVEP [22], [27]–[29]. Therefore, we propose the use of the complex spectrum features directly as input to the classifier.

2.2.3 Complex Spectrum Features. The magnitude and phase related information can be extracted from the FFT of a signal. The input time-domain signal was transformed into the complex FFT representation using the standard FFT computation with a resolution of 0.2930Hz. Next, the frequency components of the real part and the imaginary part along each channel were extracted between 3Hz and 35Hz

resulting in two vectors of length 110. These two vectors were concatenated into a single feature vector as $I = Re(X)||Im(X)$, where the first halve contained the real part and the second halve contained the imaginary part of the complex FFT. The resultant signal was stacked one below the other to form a matrix with dimensions $N_{ch} \times N_{fc}$, where $N_{fc} = 220$. This approach of using the complex FFT as input to the CNN is referred to as the C-CNN method. An example of the input I_{C-CNN} for Dataset 1 is defined as (2):

$$I_{C-CNN} = \begin{bmatrix} Re\{FFT(x_{O1})\}, Im\{FFT(x_{O1})\} \\ Re\{FFT(x_{Oz})\}, Im\{FFT(x_{Oz})\} \\ Re\{FFT(x_{O2})\}, Im\{FFT(x_{O2})\} \end{bmatrix} \quad (2)$$

2.3 Convolutional Neural Network

The CNN architecture used in this study was the one proposed in our previous study [32] and was inspired by the one proposed in [21]. Figure 2 illustrates the CNN architecture used in this study. The CNN consists of four main layers, an input layer, two convolutional layers, and a fully connected output layer. The features extracted in the previous step were provided as input to the CNN. The input layer of the CNN had dimensions $N_{ch} \times N_{fc}$. This was followed by the convolutional layer Conv_1 which was designed based on the intuition of spatial filtering. This layer performed 1D convolutions across the channel dimension (N_{ch}) with kernel dimensions of $N_{ch} \times 1$. The objective for this layer was to learn to weight the contribution of each channel differently. The number of feature maps in the Conv_1 layer was $2 * N_{ch}$ and each feature map had dimensions $1 \times N_{fc}$. The Conv_2 layer operated on the spectral representation of the input. The kernel dimension for this layer was 1×10. The number of feature maps in this layer were $2 * N_{ch}$. As a result of the convolution, the feature maps in this layer had the dimensions equal to $1 \times (N_{fc} - 10 + 1)$. Batch normalization was performed on the outputs of layers Conv_1 and Conv_2. The rectified linear unit (ReLU) was used as the activation function. Dropout was added to the network as a regularization technique to prevent overfitting. Batch normalization was shown to reduce the internal covariance within input samples resulting in the samples having zero mean and unit variance [37]. Dropout and batch normalization were shown to improve the generalization performance and training speed of neural networks [24], [37]. The output layer of the network consisted of K units equal to the number of SSVEP classes in the input data. The output layer was equipped with the *softmax* function to output the probability that a given input segment belonged to a particular class.

2.3.1 Training Parameters. The weights of the CNN were initialized based on a Gaussian distribution $\sim N(0,0.01)$. The network was trained using the backpropagation technique by minimizing the categorical cross-entropy loss function. The stochastic gradient descent with momentum was used as the optimization algorithm for training the network. A grid search was employed as the search strategy for hyper parameters to find the best training values for these parameters. The search space was defined as follows: Learning Rate (α) : $\{0.001, 0.002, 0.005, 0.01, 0.1\}$, Mini Batch size (B) : $2^b;\ b \in \{5, 6, 7, 8, 9, 10\}$, Dropout Ratio (D) : $\{0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, L2 Regularization (L) : $\{0.0001, 0.0005, 0.001, 0.005\}$, Number of Epochs (E) : $\{20, 30, 40, 50, 60\}$, and the ones that led to the best average accuracy across all participants were chosen. The hyper-parameter optimization was performed for four combinations of dataset (dataset 1 and 2) and pipelines (M-CNN and C-CNN), separately. Within each of the four combinations, the same hyper-parameters were used for all participants and window sizes.

2.3.2 User-Dependent Training Procedure (UD). In this method, a classifier was trained using the data of one single participant and the classifier was validated on the same participant's data. To achieve this, 10-fold cross-validation was performed on each participant's dataset. First, all trials of one participant were pre-processed using different window lengths (W) and both types of features were extracted. The pre-processed trials epochs were split into ten non-overlapping parts and the CNN was trained separately for each window length on nine parts and tested on the one remaining part. This procedure was carried out for Dataset 1. For Dataset 2, similar 10-fold cross-validation was performed on the 1s window

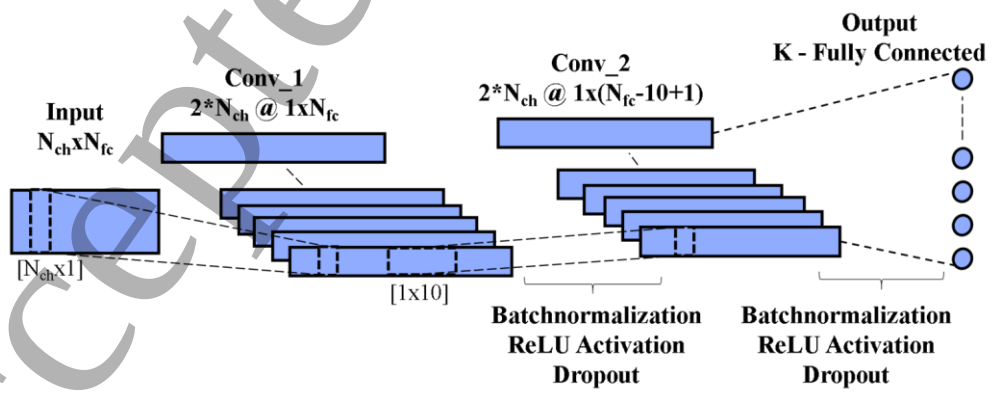


Figure 2. Convolutional Neural Network Architecture

length of the data. No other window length was used because 1s was the window length used in [12], [22] for comparison. The methods using this type of training and magnitude spectrum features were referred to as UD-M-CNN and using complex spectrum features were called UD-C-CNN. The total number of 1 s segments in the training fold were: 2470 (Dataset 1) and 648 (Dataset 2) and testing fold were 274 (Dataset 1) and 72 (Dataset 2) respectively. The final parameters of the network were chosen as: $\alpha = 0.001$, $momentum = 0.9$, $D = 0.25$, $L = 0.0001$, $E = 40$, $B = 256$ (Dataset 1), and $E = 50$, $B = 64$ (Dataset 2). An example of the UD M-CNN and C-CNN implementation on Dataset 2 in Python has been made publicly available*.

2.3.3 User-Independent Training Procedure (UI). The proposed method was evaluated in a UI training scenario for its efficacy to classify novel unseen user's SSVEP data, leading to a calibration-free system. In this method, a leave-one-participant-out method was used for training and validation of the classifier. If a given dataset contains P participants, then the classifier was trained by combining the data of P-1 participants and tested on the data of the single unseen participant. This procedure was performed individually for each feature extraction method and for each window length of data. For example, the total number of 1 s segments in training fold were: 54880 (Dataset 1) and 6480 (Dataset 2) and testing fold were 2744 (Dataset 1) and 720 (Dataset 2) respectively. The parameters that resulted in average highest accuracy across all participants were selected. The methods using this type of training with the two types of feature extraction methods were referred to as UI-M-CNN and UI-C-CNN respectively. The final parameters of the network for Dataset 1 were chosen as: $\alpha = 0.001$, $momentum = 0.9$, $L = 0.0001$, $D = 0.25$, $E = 50$, $B = 1024$ (C-CNN), and $B = 512$ (M-CNN). For Dataset 2, $\alpha = 0.001$, $momentum = 0.9$, $D = 0.25$, $E = 50$, $B = 256$, $L = 0.001$ (M-CNN) and $L = 0.005$ (C-CNN).

2.4 Canonical Correlation Analysis (CCA)

CCA was performed on each segment of the EEG data. It is a multivariate statistical method used to find the underlying correlation between two sets of multidimensional variables. Prior studies have shown that CCA can produce superior performance in detecting SSVEP responses in EEG [7], [8]. And most widely used as a baseline classification method for SSVEP detection [6], [12], [13], [27]. CCA is based on linear transformations. Consider the transformations $x = X^T w_x$ and $y = Y^T w_y$, where X refers to the set of multi-channel EEG data and Y refers to a set of reference signals of the same length as X. The objective of CCA was to find projection vectors w_x and w_y that maximize the correlation between x and y by solving the following:

$$\rho(x,y) = max_{w_x,w_y} \frac{\mathrm{E}\left[w_x^T X Y^T w_y\right]}{\sqrt{\mathrm{E}[w_x^T X X^T w_x]\mathrm{E}\left[w_y^T Y Y^T w_y\right]}}. \quad (3)$$

$$Y_n = \begin{bmatrix} sin(2\pi f_n t) \\ cos(2\pi f_n t) \\ . \\ . \\ sin(2\pi N_h f_n t) \\ cos(2\pi N_h f_n t) \end{bmatrix}, t = \left[\frac{1}{f_s}, \frac{2}{f_s}, \ldots, \frac{N_s}{f_s}\right], \quad (4)$$

The maximum of ρ with respect to w_x and w_y was the maximum correlation. The reference signals Y_n were defined as (4), where $Y_n \in \mathbb{R}^{2N_h \times N_s}$, f_n was the stimulation frequency, f_s was the sampling frequency, N_s was the number of samples, and N_h was the number of harmonics. In this study, $N_h = 2$. The canonical features ρ_{fi}, where i =1, 2, …, K were extracted for each segment of the EEG data, and the output class C for a given sample was determined as: $C = argmax\ (\rho_{fi})$.

2.5 Filter-Bank Canonical Correlation Analysis (FBCCA)

FBCCA is a user-independent variant of the CCA method [6], [15]. In this method, the multi-channel EEG data X was decomposed into J sub-band components ($X_j,\ j = 1, 2, \ldots, J$) and the standard CCA was applied to each of the sub-band components separately. The correlation values between the sub-band component X_j and the predefined reference signals Y_n belonging to the i^{th} stimulation frequency, denoted by $\rho_{j,i}$ was calculated. A weighted sum of squares of the correlation values were calculated as the feature vector for SSVEP detection:

$$\tilde{\rho}_i = \sum_{j=1}^{J} w(j)\rho_{j,i}^2, \quad (5)$$

where j is the index of the sub-band. The weight vector corresponding to the sub-band components was defined as:

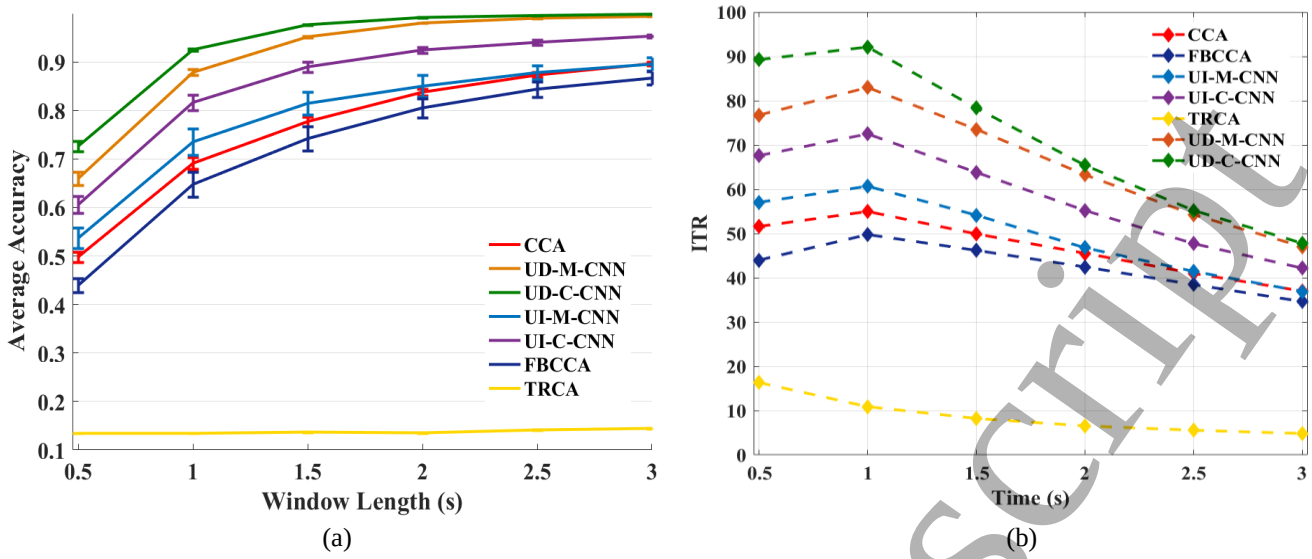
$$w(j) = j^{-a} + b, j \in [1,\ J] \quad (6)$$

where a and b are constants that maximize the classification performance. In this study, the following values were set empirically, $J = 5$, $N_h = 2$, $a = 1.25$ and $b = 0.25$. The 5 different sub-bands with the low cut-off and high cut-off frequencies for Dataset 1 were designed as (Hz): (6, 40), (10, 40), (14, 40), (20, 40) and (26, 40) and for Dataset 2 were: (Hz) (6.5, 80), (12.5, 80), (18.5, 80), (24.5, 80) and (30.5, 80).

2.6 Task-Related Component Analysis (TRCA)

TRCA is a user-dependent training method used to obtain spatial filters that extract task related source activities from multi-channel EEG data [14]. Using individual training data, TRCA extracts task related components by maximizing their reproducibility during task periods. Consider the multi-channel

*Source Code: https://github.com/aaravindravi/Brain-computer-interfaces



(a) (b)

Figure 3. Performance comparison of Dataset 1 – (a) Comparison of the average accuracies, (b) comparison of the average ITR (bits/min), across all participants for the different classification methods for data lengths of W = {0.5s, 1s, 1.5s, 2s, 2.5s, 3s}. Chance level = 0.14. The vertical bars indicate the standard deviation among the participants at each W.

EEG $x(t) \in \mathbb{R}^{N_c}$, TRCA finds a linear coefficient vector $w \in \mathbb{R}^{N_c}$ to maximize the inter-trial correlation of its projections $y(t) = w^T x(t)$, which is called a task-related component. The h[th] trial in the observed EEG is given by $x^{(h)} \in \mathbb{R}^{N_c \times N_s}$ and the task-related component is given by $y^{(h)} \in \mathbb{R}^{N_s}$. The covariance $C_{h1,h2}$ between the h_1 [th] and h_2 [th] trials of $y^{(h)}$ is described as:

$$C_{h1,h2} = Cov\left(y^{(h1)},\ y^{(h2)}\right) = \sum_{j1,j2=1}^{N_c} w_{j1} w_{j2} Cov\left(x_{j1}^{(h1)},\ x_{j2}^{(h2)}\right), \quad (7)$$

All possible combinations of N_t trials are summed as:

$$\sum_{\substack{h1,h2=1 \\ h1 \neq h2}}^{N_t} \left(C_{h1,h2}\right) = \sum_{\substack{h1,h2=1 \\ h1 \neq h2}}^{N_t} \sum_{j1,j2=1}^{N_c} w_{j1} w_{j2} Cov\left(x_{j1}^{(h1)}, x_{j2}^{(h2)}\right) = w^T S\, w. \quad (8)$$

To obtain a finite solution, the variance of $y(t)$ is constrained as:

$$Var(y(t)) = \sum_{j1,j2=1}^{N_c} w_{j1} w_{j2} Cov\left(x_{j1}^{(h1)}, x_{j2}^{(h2)}\right) = w^T Q\, w = 1. \quad (9)$$

The constrained optimization problem can then be solved by:

$$\hat{w} = \underset{w}{\operatorname{argmax}} \frac{w^T S\, w}{w^T Q\, w} \quad (10)$$

The eigenvalues of the matrix $Q^{-1}S$ indicate the task consistency among multiple trials. The eigenvector corresponding to the largest eigenvalue $\hat{w}$ was selected as the spatial filter to extract task related components. In this study, the following values were set empirically, $N_h = 2$, $a = 1.25$ and $b = 0.25$. The number of sub-bands was set as $J = 5$ and the same sub-bands used in FBCCA were used in this method. The performance of this method was evaluated on the segmented SSVEP data of each participant based on a 10-fold cross-validation scheme.

2.7 Statistical Analysis

Statistical analysis was performed on the results of both the datasets to evaluate the performance of the different classification methods. The user-dependent training methods and user independent training methods were compared with each other and with the baseline CCA method. Additional comparisons were performed with FBCCA and TRCA. A mixed-effect model ANOVA was used to evaluate the classification methods. The metric of interest was the overall accuracy of each method in classifying the different SSVEP targets. Therefore, the response variable was the classification accuracy. The participant was a random factor, the window length (W) was a random factor with six levels (W=[0.5s, 3s]), and the classification algorithm was a fixed factor with seven levels (CCA, FBCCA, TRCA, UD-M-CNN, UD-C-CNN, UI-M-CNN, UI-C-CNN) respectively (Dataset 1). The null hypothesis was that the classification accuracy was same for all classification algorithms. A 95% confidence interval was used for the comparison and analysis. The same statistical analysis was performed on both datasets with slight modifications for Dataset 2, in which window length was not a factor as it was fixed as W=1s.

3. Results

3.1 Dataset 1

Figure 3(a) illustrates the average classification accuracies of all the methods across 21 participants for Dataset 1 at different window lengths. Across all windows lengths, the classification methods can be ranked from highest to lowest as follows: UD-C-CNN, UD-M-CNN, UI-C-CNN, UI-M-CNN, CCA, FBCCA and TRCA. Among the UI methods, the UI-C-CNN achieved higher performance than UI-M-CNN, CCA and FBCCA. Similarly, among the UD methods, the UD-C-CNN achieved higher performance than UD-M-CNN, CCA and TRCA. Compared to all other methods, the TRCA provided the lowest performance close to chance level performance across all windows. This method was not included in the statistical analysis. The mixed-effect model ANOVA revealed a significant effect of classification methods (p<0.001). Post-hoc comparisons with Bonferroni simultaneous comparison indicated that there was a significant improvement in performance using UD-M-CNN, UD-C-CNN, and UI-C-CNN when compared to CCA (p<0.001) and FBCCA (p<0.001). The UI-M-CNN performed significantly better than FBCCA (p=0.002). There was no significant difference between UI-M-CNN vs. CCA (p>0.05) and FBCCA vs. CCA (p=0.153). Further analysis was carried out to compare between the UD and UI CNN methods based on each feature extraction technique and all comparisons were statistically significant: UI-M-CNN vs. UD-M-CNN (p<0.001), UI-C-CNN vs. UD-C-CNN (p<0.001), UD-M-CNN vs. UI-C-CNN (p=0.009), UD-C-CNN vs. UI-M-CNN (p<0.001). These results indicate that the proposed C-CNN pipeline outperformed the M-CNN pipeline when both methods were applied either in UI or UD scheme. Even when the C-CNN was used in the UI scheme, it performed similarly to M-CNN used in the UD scheme, highlighting its advantages.

Subsequent analysis was performed to measure the interactions between window lengths and the different classification methods. The tests revealed that both UD CNN methods outperformed CCA (p<=0.001) and FBCCA (p<0.001) across all window lengths. Among the UI methods, UI-C-CNN had significant improvement than CCA for window lengths between 0.5s and 2s (0.001<=p<0.022). Across all windows, UI-C-CNN was significantly better than FBCCA (p<=0.003). Similarly across all windows, UD-M-CNN was significantly better than UI-M-CNN (p<=0.002). There was a significant difference in accuracy at lower windows from 0.5s – 1.5s between UD-C-CNN and UI-C-CNN (0.002<=p<0.043). The average accuracies of the different methods for 1s window length were: TRCA: 13.4±1.5%, FBCCA: 64.8±15.6%, CCA: 69.1±10.8%, UI-M-CNN: 73.5±16.1%, UI-C-CNN: 81.6±12.3%, UD-M-CNN: 87.8±7.6% and UD-C-CNN: 92.5±5%. Figure 3(b) illustrates the average information transfer rate (ITR) for all methods across different window lengths calculated with 0.5s gaze-shift period. The maximum ITR obtained for W=1s were: TRCA: 10.9 bits/min, FBCCA: 49.8 bits/min, CCA: 55 bits/min, UI-M-CNN: 60.7 bits/min, UI-C-CNN: 72.5 bits/min, UD-M-CNN: 83 bits/min and UD-C-CNN: 92.1 bits/min. It can be inferred that both the UD training-based CNN methods have outperformed the UI training based methods and CCA.

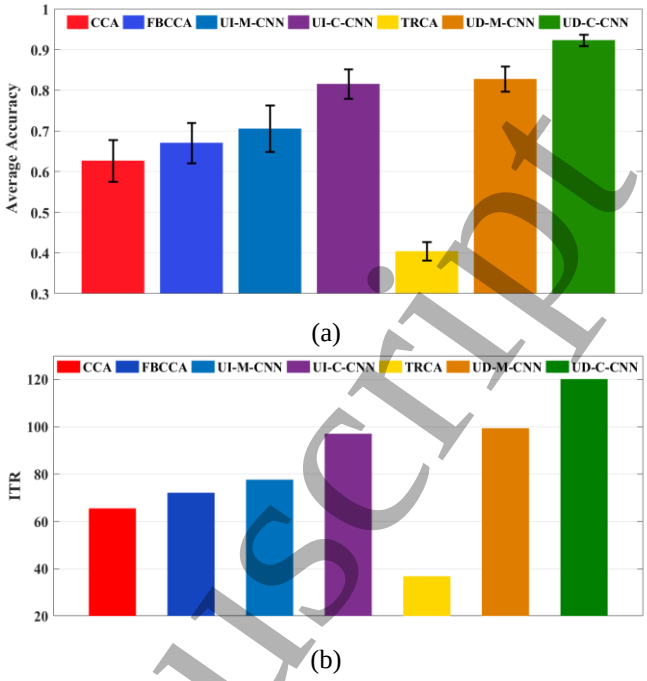


Figure 4. The performance comparison of Dataset 2 – (a) Comparison of the average accuracies, (b) comparison of the average ITR (bits/min) across all participants for the different classification methods for data lengths of W = 1s. Chance level = 0.083. The vertical line overlaying each bar indicates the standard deviation among all participants in each method.

3.2 Dataset 2

Figure 4(a) summarizes the average accuracies of all the classification methods for Dataset 2 across 10 participants for the data length of 1 s. It can be inferred from the figure that the UD-C-CNN method achieves the highest accuracy of 92.33±11.1%. Among the UD and UI methods, the UD CNN methods outperform the UI CNN methods, FBCCA and CCA, as expected. The TRCA method had the lowest performance among all other methods and was not included in subsequent analysis. The likely explanation for the poor performance of TRCA is that the method is not suitable for asynchronously processed SSVEP data, explained further in subsequent sections. The average accuracies of the different methods for data length of 1s were: UD-C-CNN: 92.33±11.1%, UD-M-CNN: 82.77± 16.7%, UI-C-CNN: 81.6±18%, UI-M-CNN: 70.5±22%, FBCCA: 67.1±21%, CCA: 62.7±21.5% and TRCA: 40.4±14%.

The mixed-effect model ANOVA revealed a significant difference between all the classification methods (p<0.001). Post-hoc Bonferroni simultaneous comparison was performed to compare the different algorithms. The UD-C-CNN, UD-M-CNN and UI-C-CNN significantly outperformed CCA (p<0.001) and FBCCA (p<0.002). There was a significant

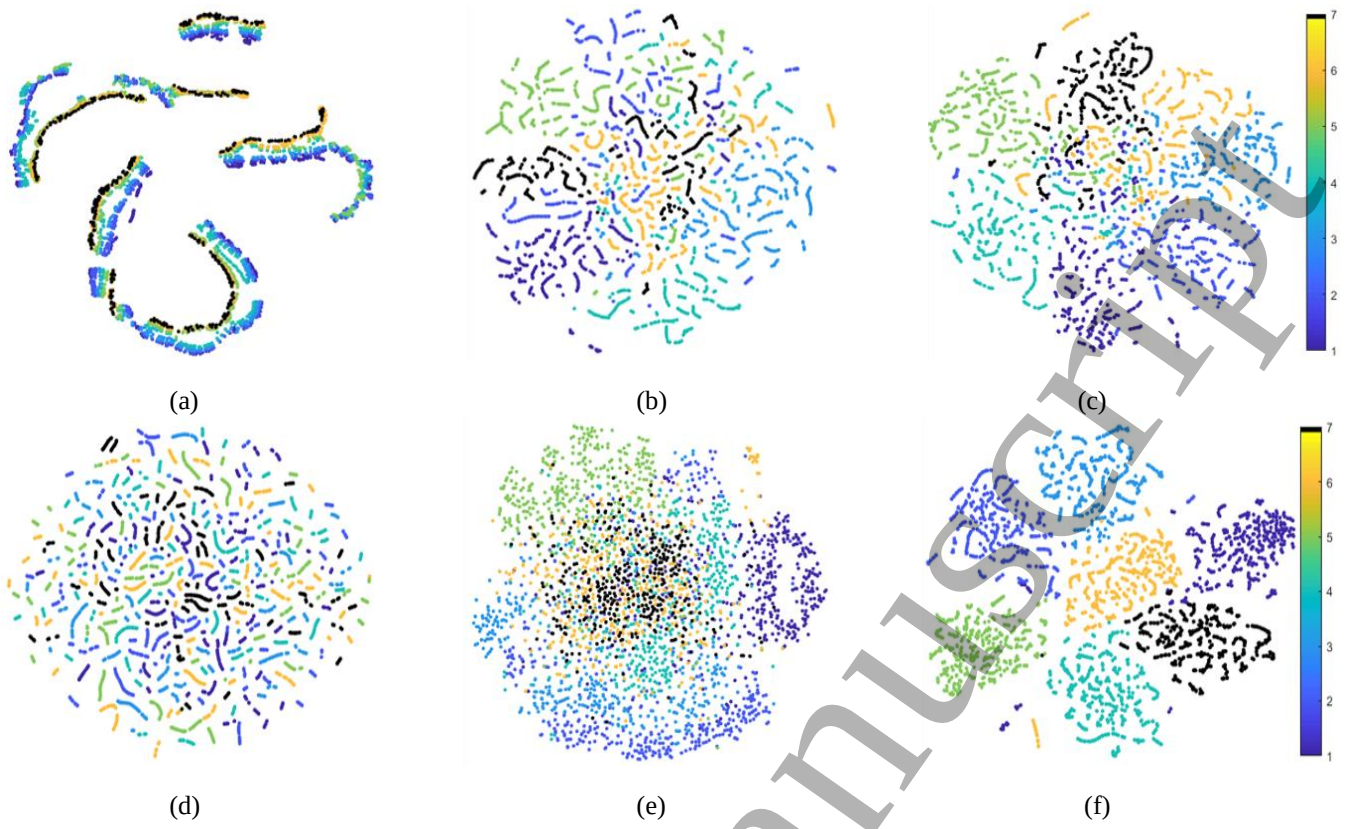


Figure 5. Dataset 1 - (a-c) Feature Visualization of an unseen participant using t-SNE – UI-M-CNN. (d-f) Feature Visualization of an unseen participant using t-SNE – UI-C-CNN. (a) Input magnitude spectrum features. (b) Output of Conv_1_ReLU Layer of M- CNN. (d) Input complex spectrum features. (e) Output of Conv_1_ReLU Layer of C-CNN. (f) Output of the Conv_2_ReLU layer of C-CNN.

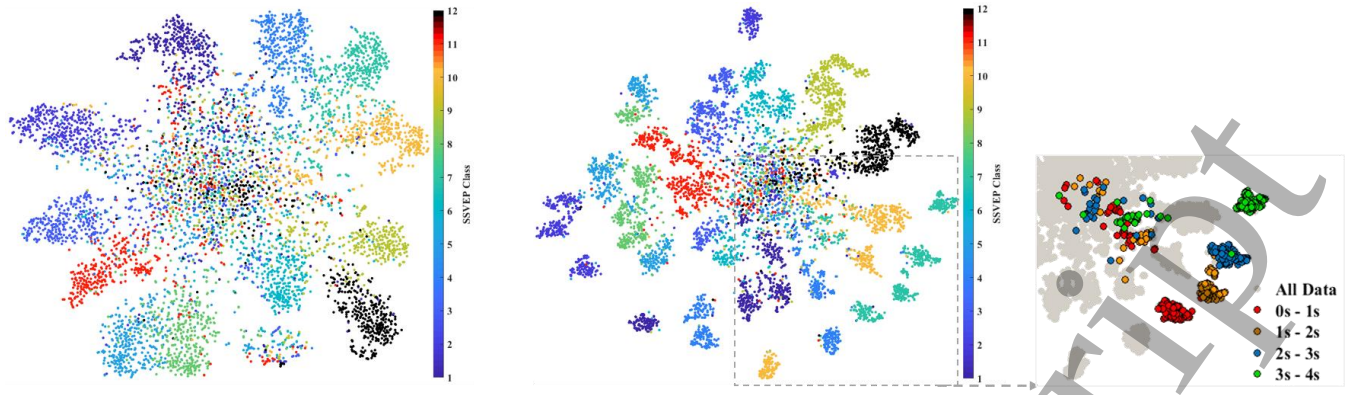
difference between UI-M-CNN and UI-C-CNN (p=0.037). There was no significant difference between UI-M-CNN vs. FBCCA (p>0.05) and UI-M-CNN vs. CCA (p=0.398). There was no significant between UD-M-CNN and UD-C-CNN (p=0.118). Further analysis was carried out to compare between the UD and UI CNN methods based on each feature extraction technique: UI-M-CNN vs. UD-M-CNN (p=0.014), UI-C-CNN vs. UD-C-CNN (p=0.045), and UD-C-CNN vs. UI-M-CNN (p<0.001). The difference between UD-M-CNN vs. UI-C-CNN (p=1) was not significant. Figure 4(b) illustrates the average ITR for all the methods for a W=1s calculated with 0.5s of gaze-shifting as: TRCA: 36.8 bits/min, CCA: 65.5 bits/min, FBCCA: 72 bits/min, UI-M-CNN: 77.6 bits/min, UI-C-CNN: 97 bits/min, UD-M-CNN: 99.3 bits/min and UD-C-CNN: 120.1 bits/min.

3.3 Computational Load Analysis

A computational load analysis was performed on both types of CNN models. The CNN pipelines were implemented using the MATLAB Deep Learning Toolbox and were trained on an Intel Core i5-8400 CPU @ 2.80 GHz and 8 GB RAM. The total number of trainable parameters for the Dataset 1 were: UD-M-CNN = UI-M-CNN = 4663 and UD-C-CNN = UI-C-CNN = 9283. The overall training time to train 1 s segments were: UD-M-CNN: 6 seconds, UD-C-CNN: 12 seconds, UI-M-CNN: 3 minutes 20 seconds and UI-C-CNN: 7 minutes 17 seconds. For Dataset 2, the total number of trainable parameters were: UD-M-CNN = UI-M-CNN = 22188 and for UD-C-CNN = UI-C-CNN = 43308. The overall training time to train 1s segments were: UD-M-CNN: 6 seconds, UD-C-CNN: 10 seconds, UI-M-CNN: 53 seconds and UI-C-CNN: 1 minute 50 seconds; the number of training samples was 6480.

4. Discussions

The results of this study clearly indicate that the UD method performs better than UI methods. It is interesting to note that from the results from both datasets, the C-CNN methods outperformed the other methods in both UD and UI training scenarios. The UI-C-CNN based method performed similar to the UD-M-CNN method and outperformed the UI-M-CNN method. Further investigation was performed to compare the results of the UI-M-CNN with the UI-C-CNN by visualizing the learned feature representations of the CNN on both the datasets. The features were visualized using the t-Stochastic Neighborhood Embedding (t-SNE) technique [38]. This method is a widely used feature visualization technique which enables us to visualize high-dimensional feature spaces in 2 or 3 dimensions [16], [22], [39]. The features from the input layer,



(a) (b) (c)

Figure 6. Dataset 2 - Feature Visualization of all participants using t-SNE. (a) Output of the Conv_2_ReLU layer of UI-M-CNN. (b) Output of the Conv_2_ReLU layer of UI-C-CNN (Right), (c) Segment level clustering for SSVEP Class 12.25 Hz of UI-C-CNN.

the ReLU layer of Conv_1 and the ReLU layer of Conv_2 were extracted. The 1 s long SSVEP segments were visualized for both datasets using the magnitude and complex features for the UI method.

4.1 Dataset 1

Figure 5 (a-f) illustrates the features at different layers of the network for the UI-M-CNN (a-c) and UI-C-CNN (d-f) methods respectively. To achieve this, the CNN was trained on the P-1 participants' data and the unseen test participant's data was forward propagated into the pre-trained network and features were extracted at the output of each layer. Each data point in the figure belongs to 1 s segments of a single trial and is colored based on the class label. It can be observed that the features become more and more clustered as we progress into the deeper layers of the network for both methods. Comparing the outputs of the ReLU of Conv_2 layers of the M-CNN (c) and C-CNN (f), it can be observed that using the complex representation of the inputs leads to better clustering and class separation. The CNN has learned seven unique classes from the training data and is able to cluster the unseen participant's data into one of the seven classes. There is smaller overlap between the classes in the C-CNN compared to the M-CNN. This is also evident from the classification accuracies for all window lengths in which the C-CNN outperforms the M-CNN method. Therefore, by including the real and imaginary parts of the complex FFT as input to the CNN, we observe that the CNN can extract significantly more discriminative features that lead to better overall separation and classification accuracy when compared to the magnitude spectrum features.

4.2 Dataset 2

Similar feature visualization was performed on Dataset 2. Figure 6(a) illustrates the overall feature clustering on Dataset 2 of all the participants. The UI-M-CNN and UI-C-CNN methods are compared based on the t-SNE visualization. It can be clearly seen that the clustering in the feature space of the ReLU of the Conv_2 layer of C-CNN shows distinct clusters compared to M-CNN. This type of class separation aids in achieving better classification accuracy. A previous CNN study in [22] used time domain features on this dataset and showed that within-class clusters were captured by their proposed method. In the present study, using complex spectrum features and a lighter CNN architecture, similar within-class differences, similar to the ones reported in [22], have been learned by the UI-C-CNN. Figure 6(b) illustrates an example of the trials belonging to the 12.25 Hz class. Four distinct clusters can be identified in this class. Further analysis found that these actually correspond to the four non-overlapping 1s segments of the 4s trials of the 12.25 Hz data. The clusters were colour-coded according to the segment label in Figure 6(c). It can be observed that there was a segment level clustering that the CNN has learned, i.e. all the 1s segments were clustered into four groups as follows: 0s - 1s, 1s - 2s, 2s - 3s and 3s - 4s. This separation could be due to the fact that there exists phase related information that has been extracted from the complex representation of the input and this has enabled in clustering into four clusters. And such level of detailed discriminative information was not presented with the M-CNN method. This shows evidence that the UI-C-CNN method is capable of extracting phase and amplitude related features. These clustering results are consistent with the findings reported in [22] where a radial phase plot analysis on the 1s segments showed that segments between 0s - 1s, 1s - 2s had separable phases with the segments between 2s - 3s, 3s - 4s and the clustering revealed that the 1st and 2nd segments were on opposite phases of the 3rd and 4th segments of the trial. These results were also consistent across multiple classes which can be observed in Figure 6(b) of the present study. From these results, it is evident that the C-CNN method proposed in this study can improve the overall SSVEP decoding performance significantly.

The results obtained on both the datasets indicate that the TRCA method had the lowest performance among all compared methods. One of the likely reasons for this is that, in the current study the SSVEP data was processed in an

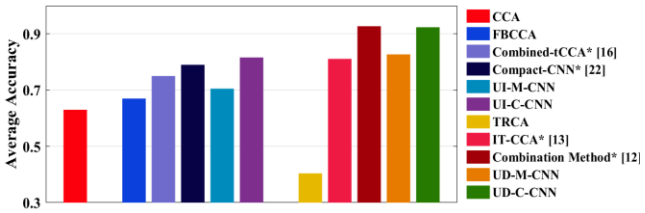


Figure 7. Comparing the UD and UI methods on Dataset 2 for 1s window length with other methods as reported in the literature. *Values used directly from the respective studies.

asynchronous manner in which the training trials were segmented based on a fixed window and step size, and the data was not phase locked. Previous studies applying TRCA were based on synchronous SSVEP paradigms with fixed windows of data which were always tied precisely to the onset of the stimulus. A similar observation was reported by [22] where the combined-CCA method performed poorly for asynchronously processed SSVEP data. Therefore, future studies could investigate the application of TRCA to asynchronous SSVEP paradigms.

Further analysis was performed to compare the results achieved on the Dataset 2 with the ones reported in the literature. The methods presented in the present study were CCA, FBCCA, TRCA, UD-M-CNN, UD-C-CNN, UI-M-CNN, and UI-C-CNN. In a recent study, it was reported that a vast majority of published studies based on deep learning for EEG based BCIs did not compare the proposed techniques to state-of-the-art methods or they performed biased comparisons [40]. In the current study, we have attempted to provide a comparison of our methods with other techniques proposed in the literature in an unbiased way. Therefore we compared our methods with two UD and two UI methods as identified in [6]. The following methods that were selected were those published studies that tested on Dataset 2 and reported the results for the 1 s data length. We directly compared the accuracies reported in these published studies with the methods evaluated in this presented paper. Among UD methods, the combination method [12] and Independent Template based CCA (IT-CCA) [13] were selected. Among the UI methods, the Compact-CNN [22] and the Combined-tCCA [16] methods were selected. Figure 7 compares the classification accuracies of the calibration-free CCA, FBCCA, TRCA, UD and UI training methods of CNN with the accuracies reported by previously published studies in the literature such as [12] [13] [16] [22]. Overall results show that the UD methods achieve higher accuracies compared to UI methods and CCA. Among the UD methods, the proposed UD-C-CNN (92.33±11.1%) outperforms CCA (62.7±21.5%), UD-M-CNN (82.77±16.7%), IT-CCA (81.17±18.84%) and TRCA (40.4±14%) but it is similar in performance to the Combination method (92.78±10.22). Among the UI methods, the proposed UI-C-CNN (81.6±18%) achieves the highest performance compared to CCA (62.7±21.5%), UI-M-CNN (70.5±22%), FBCCA (67.1±21%), Compact-CNN (79±15%) and Combined-tCCA (75±24%). Only those studies that have used Dataset 2 for benchmarking have been compared here. The pooled transfer based methods were used in this comparison, and adaptive learning method such as the adaptive combined CCA [16] was not used in this comparison.

5. Conclusions and Future Work

In this study, we investigated CNN for SSVEP classification based on both the UI and UD schemes. We introduced a method to extract complex spectrum based features from SSVEP and provided as input to a CNN for classification. The classifier was evaluated in both UD and UI training schemes. The proposed method was compared with the magnitude spectrum features (M-CNN) and CCA. The results indicated that the proposed C-CNN outperformed both M-CNN and CCA across all processing window lengths in both UI and UD training scenarios. The UD based training methods consistently achieved higher classification accuracies compared to the UI methods, as one would expect. The UD-C-CNN based method ranked highest among the compared methods. Within the UI methods, the UI-C-CNN achieved the highest performance. Further, its performance was similar to UD-M-CNN. Visualizing the features extracted by the UI-C-CNN method indicated that the method likely learned phase related information from the SSVEP data. The proposed methods and comparisons performed on a publicly available twelve class SSVEP dataset showed that the findings were consistent with the ones reported in the literature. The UI-C-CNN method achieved the highest accuracy among most tested UI methods on the public dataset, and the UD-C-CNN performed similarly to the combined method, which was the best SSVEP decoder in [12].

A comparative study was required to inform whether the cost of training would be borne by the user (in case of UD training) or by the developer of the BCI (in case of UI training) [6]. We have addressed some of the points in this study. There is a trade-off between achieving high classification accuracy versus the cost of collecting training data. If the performance of the system was of higher priority, UD methods offer the best accuracy compared to UI and training-free methods. But this will require each user to undergo a calibration session, which could lead to issues such as poor user-compliance. On the other hand, if the developers are willing to collect training data from multiple participants, then the UI-C-CNN method proposed in this study offers a good balance between performance and cost of training data. Transfer learning based method has the potential to provide the combined advantage of high accuracy of the UD method and the training-free UI method. A future study can explore using multiple participants' data to build a model and fine-tune the model by collecting minimal calibration data from the unseen user. With pre-trained models, online adaptation strategies can be employed for improving the overall performance of the BCI. Number of participants required to build a sufficiently accurate UI model should be explored. In the current study, we evaluated the methods on two datasets consisting of 21 participants and 10 participants each. The methods presented in this study have been evaluated in an offline manner, therefore future studies can explore online performance in an asynchronous SSVEP-BCI setting.

In conclusion, the proposed C-CNN based methods are suitable candidates for SSVEP-based BCIs and provide improved performance in both user-dependent and user-independent training scenarios.

Acknowledgments

We would like to acknowledge support from NSERC – CREATE, Training in Global Biomedical Technology Research and Innovation at the University of Waterloo. [CREATE Funding 401207296]. We thank all the participants who took part in the study. We thank the anonymous reviewers for their valuable feedback and suggestions.

References

[1] G. Dornhege, J. R. Millán, T. Hinterberger, D. Mcfarland, K. Müller, and A. B. Book, "Towards Brain-Computer Interfacing," pp. 31–42, 2007.

[2] B. Z. Allison, D. J. McFarland, G. Schalk, S. D. Zheng, M. M. Jackson, and J. R. Wolpaw, "Towards an independent brain-computer interface using steady state visual evoked potentials," *Clin. Neurophysiol.*, vol. 119, no. 2, pp. 399–408, 2008.

[3] D. Lesenfants *et al.*, "An independent SSVEP-based brain-computer interface in locked-in syndrome," *J. Neural Eng.*, vol. 11, no. 3, p. 035002, 2014.

[4] X. Zhang, G. Xu, J. Xie, M. Li, W. Pei, and J. Zhang, "An EEG-driven Lower Limb Rehabilitation Training System for Active and Passive Co-stimulation," *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS*, vol. 2015–Novem, no. August, pp. 4582–4585, 2015.

[5] Y. Jiang, X. Zhao, R. Abiri, S. Borhani, and E. W. Sellers, "A comprehensive review of EEG-based brain–computer interface paradigms," *J. Neural Eng.*, vol. 16, no. 1, p. 011001, 2018.

[6] R. Zerafa, T. Camilleri, O. Falzon, and K. P. Camilleri, "To train or not to train? A survey on training of feature extraction methods for SSVEP-based BCIs," *J. Neural Eng.*, vol. 15, no. 5, p. 051001, 2018.

[7] Z. Lin, C. Zhang, W. Wu, and X. Gao, "Frequency recognition based on canonical correlation analysis for SSVEP-Based BCIs," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 6, pp. 1172–1176, 2007.

[8] G. Bin, X. Gao, Z. Yan, B. Hong, and S. Gao, "An online multi-channel SSVEP-based brain-computer interface using a canonical correlation analysis method," *J. Neural Eng.*, vol. 6, no. 4, p. 046002, 2009.

[9] H. Cecotti, "A self-paced and calibration-less SSVEP-based brain-computer interface speller," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 18, no. 2, pp. 127–133, 2010.

[10] H. Cecotti and D. Coyle, "Calibration-less detection of steady-state visual evoked potentials-comparisons and combinations of methods," *Proc. Int. Jt. Conf. Neural Networks*, pp. 4050–4055, 2014.

[11] I. Volosyak, D. Valbuena, T. Lüth, T. Malechka, and A. Gräser, "BCI demographics II: How many (and What Kinds of) people can use a high-frequency SSVEP BCI?," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 3, pp. 232–239, 2011.

[12] M. Nakanishi, Y. Wang, Y. Te Wang, and T. P. Jung, "A comparison study of canonical correlation analysis based methods for detecting steady-state visual evoked potentials," *PLoS One*, vol. 10, no. 10, pp. 1–18, 2015.

[13] Y. Wang, M. Nakanishi, Y. Te Wang, and T. P. Jung, "Enhancing detection of steady state visual evoked potentials using individual training data," *2014 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC 2014*, pp. 3037–3040, 2014.

[14] M. Nakanishi, Y. Wang, X. Chen, Y. Te Wang, X. Gao, and T. P. Jung, "Enhancing detection of SSVEPs for a high-speed brain speller using task-related component analysis," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 1, pp. 104–112, 2018.

[15] X. Chen, Y. Wang, S. Gao, T. P. Jung, and X. Gao, "Filter bank canonical correlation analysis for implementing a high-speed SSVEP-based brain-computer interface," *J. Neural Eng.*, vol. 12, no. 4, p. 046008, 2015.

[16] N. R. Waytowich, J. Faller, J. O. Garcia, J. M. Vettel, and P. Sajda, "Unsupervised adaptive transfer learning for Steady-State Visual Evoked Potential brain-computer interfaces," in *2016 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2016*, 2016, pp. 4135–4140.

[17] O. Faust, Y. Hagiwara, T. J. Hong, O. S. Lih, and U. R. Acharya, "Deep learning for healthcare applications based on physiological signals: A review," *Comput. Methods Programs Biomed.*, vol. 161, pp. 1–13, 2018.

[18] Y. Roy, H. Banville, I. Albuquerque, A. Gramfort, T. H. Falk, and J. Faubert, "Deep learning-based electroencephalography analysis: A systematic review," *J. Neural Eng.*, vol. 16, no. 5, p. 051001, 2019.

[19] A. Craik, Y. He, and J. L. Contreras-Vidal, "Deep learning for electroencephalogram (EEG) classification tasks: A review," *J. Neural Eng.*, vol. 16, no. 3, p. 031001, 2019.

[20] H. Cecotti, "A time-frequency convolutional neural network for the offline classification of steady-state visual evoked potential responses," *Pattern Recognit. Lett.*, vol. 32, no. 8, pp. 1145–1153, 2011.

[21] N. S. Kwak, K. R. Müller, and S. W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PLoS One*, vol. 12, no. 2, pp. 1–20, 2017.

[22] N. Waytowich *et al.*, "Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials," *J. Neural Eng.*, vol. 15, no. 6, p. 066031, 2018.

[23] T.-H. Nguyen and W.-Y. Chung, "A Single-Channel SSVEP-Based BCI Speller using Deep Learning," *IEEE Access*, vol. 7, pp. 1–1, 2018.

[24] X. Zhang *et al.*, "A Convolutional Neural Network for the Detection of Asynchronous Steady State Motion Visual Evoked Potential," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 6, pp. 1303–1311, 2019.

[25] N. K. Nik Aznan, S. Bonner, J. Connolly, N. Al Moubayed, and T. Breckon, "On the Classification of SSVEP-Based Dry-EEG Signals via Convolutional Neural Networks," *Proc. - 2018 IEEE Int. Conf. Syst. Man, Cybern. SMC 2018*, pp. 3726–3731, 2019.

[26] Y. Zhang, J. Jin, X. Qing, B. Wang, and X. Wang, "LASSO based stimulus frequency recognition model for SSVEP BCIs," *Biomed. Signal Process. Control*, vol. 7, no. 2, pp. 104–111, 2012.

[27] X. Chen, Y. Wang, M. Nakanishi, X. Gao, T. P. Jung, and S. Gao, "High-speed spelling with a noninvasive brain-computer interface," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 44, pp. E6058–E6067, 2015.

[28] T. Kluge and M. Hartmann, "Phase coherent detection of steady-state evoked potentials: Experimental results and application to brain-computer interfaces," *Proc. 3rd Int. IEEE EMBS Conf. Neural Eng.*, pp. 425–429, 2007.

[29] J. Pan, X. Gao, F. Duan, Z. Yan, and S. Gao, "Enhancing the classification accuracy of steady-state visual evoked potential-based brain-computer interfaces using phase constrained canonical correlation analysis," *J. Neural Eng.*, vol. 8, no. 3, 2011.

[30] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces," *J. Neural Eng.*, vol. 15, no. 5, p. 056013, 2018.

[31] A. Ravi, N. Heydari, and N. Jiang, "User-Independent SSVEP BCI Using Complex FFT Features and CNN Classification," *2019 IEEE Int. Conf. Syst. Man, Cybern. SMC 2019*, pp. 4175–4180, 2019.

[32] A. Ravi, J. Manuel, N. Heydari, and N. Jiang, "A Convolutional Neural Network for Enhancing the Detection of SSVEP in the Presence of Competing Stimuli," *2019 41st Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 6323–6326, 2019.

[33] A. Ravi, S. Pearce, X. Zhang, and N. Jiang, "User-Specific Channel Selection Method to Improve SSVEP BCI Decoding Robustness Against Variable Inter-Stimulus Distance," *9th Int. IEEE EMBS Conf. Neural Eng.*, pp. 283–286, 2019.

[34] A. Duszyk *et al.*, "Towards an optimization of stimulus parameters for brain-computer interfaces based on steady state visual evoked potentials," *PLoS One*, vol. 9, no. 11, pp. 1–11, 2014.

[35] Y. Wang, R. Wang, X. Gao, B. Hong, and S. Gao, "A practical VEP-based brain-computer interface," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 14, no. 2, pp. 234–239, 2006.

[36] Y. Renard *et al.*, "OpenViBE : An Open-Source Software Platform to Design , Test , and Use Brain – Computer Interfaces in Real and Virtual Environments," *Presence*, vol. 19, no. 1, pp. 35–53, 2010.

[37] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *32nd Int. Conf. Mach. Learn.*, vol. 37, pp. 448–456, 2015.

[38] G. H. Laurens van der Maaten, "Visualizing Data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[39] D. López-Sánchez and J. M. Arrieta, Angélica G. Corchado, "Deep neural networks and transfer learning applied to multimedia web mining," *14th Int. Conf. Adv. Intell. Syst. Comput.*, vol. 620, pp. 124–131, 2018.

[40] F. Lotte *et al.*, "A Review of Classification Algorithms for EEG-based Brain-Computer Interfaces: A 10-year Update," *J. Neural Eng.*, pp. 0–20, 2018.